

# ENHANCING THE QUALITY OF DECODED QUANTIZED IMAGES

## BACKGROUND OF THE INVENTION

5 This invention relates to a system for image enhancement and, more particularly, to a system for enhancing the quality of a quantized image.

As the state of the art of digital signal technology advances, related technologies such as digital image processing has experienced a corresponding growth and benefit. For example, the development and proliferation of facsimile communications allows images to be encoded into digital signals, transmitted over conventional telephone line, and decoded into a close representation of the original images. Image data are also digitally encoded for ease of storage, modification, copying, etc. As is common experience with growing technologies, the field of digital image processing is also experiencing problems with applications in new areas.

15 Problems in the area of digital image processing relate generally to achieving a balance between acceptable image distortion and bit-depth representations. In order to increase the efficiency and therefore the usefulness of digital image decoding schemes, the coding system must provide a coded set of image data that is more efficient to store, transmit, etc., than the original image data and must reproduce a decoded image with some minimum level of quality. However, the conversion of relatively high bit rate image data to lower bit rate data virtually always entails a loss of image quality.

25 One straightforward method for digitizing an image is to create an artificial grid over the image and to assign a value to each grid space representing the color of the original image at that grid space. If the grids are made small enough and the values represent a large enough range of color, then the image may be encoded and decoded with small image quality degradation. For example, display screen images are made up of an array of pixels, i.e., picture elements. On a black and white screen, each pixel has a value of one or zero representing the on/off state of the

30

pixel. In a one-to-one bit-to-pixel coding scheme, each pixel value is represented as a 1 or as a 0 and the entire screen image is encoded. The result of the encoding is an array of binary values. To decode the image, the array values are translated into a screen image having pixels on or off in the same order in which they were originally encoded.

If the image is comprised of more than two distinct colors, then more than a 1-bit code must be used to represent the pixel values. For example, if four distinct colors are present in the image, a 2-bit binary code can represent all of the values. If the image includes 256 distinct colors, an 8-bit binary code is required to uniquely represent each of the color values. The memory requirements for such coding schemes increase as the number of distinct colors in the image increases. However, with high bit-depth representation schemes, the quality of the image that results will be good as long as the digital image transmission or recovery from storage is successful.

To reduce the size of the encoded digital image, the bit-depth representation of the image may be reduced in some manner. For example, an image with a bit-depth of 6 bits per pixel requires significantly less storage capacity and bandwidth for transmission than the same sized image with 16 bits per pixel.

Decoded images, constructed by a low bit-depth representation, generally suffer from the following types of degradations: (a) quasi-constant or slowly varying regions suffer from contouring effects and amplified granular noise, and (b) textured regions lose detail.

Contouring effects, which are the result of spatial variations, in a decoded image are generally fairly obvious to the naked eye. The contouring effects that appear in the slowly varying regions are also caused by the fact that not all of the variations in the intensity of the original image are available for the decoded image. For example, if a region of the original image included an area having 4 intensity changes therein, the decoded image might represent the area with only 2

intensities. In contrast to the contouring effects, the effect of the granular noise on the viewed image is often mitigated by the very nature of the textured regions. But it can be both amplified or suppressed due to quantization, as well as altered in spectral appearance.

5           Kundu et al., U.S. Patent No. 5,218,649, disclose an image processing technique that enhances images by reducing contouring effects. The enhancement system identifies the edge and non-edge regions in the decoded image. Different filters are applied to each of these regions and then they are combined together. A low pass filter (LPF) is used on the non-edge regions, and a high-pass enhancer is used on the edge regions. 10           Kundu et al. teaches that the contour artifacts are most visible in the non-edge areas, and the LPF will remove these edges. Unfortunately, problems arise in properly identifying the edge regions (is a steep slope an edge or a non-edge?). In addition, problems arise in setting thresholds in the segmentation process because if the contours have a high enough 15           amplitude, then they will be classified as edges in the segmentation, and thus not filtered out. Moreover, the image segmentation requires a switch, or if statement, and two full-size image buffers to store the edge map and smooth map, because the size of these regions varies from image to image, all of which is expensive and computationally intensive. 20

          Chan, U.S. Patent No. 5,651,078, discloses a system that reduces the contouring in images reproduced from compressed video signals. Chan teaches that some contours result from discrete cosine transformation (DCT) compression, which are more commonly called 25           blocking artifacts. Chan observes that this problem is most noticeable in the dark regions of image and as a result adds noise to certain DCT coefficients when the block's DC term is low (corresponding to a dark area). The resulting noise masks the contour artifacts, however, the dark areas become noisier as a result. The dark areas thus have a noise similar 30           to film-grain-like noise which may be preferable to the blocking artifacts.

What is desirable is a system for reducing contouring effects of an image. Because the system does not necessarily affect the encoding or transmission processes, it can be readily integrated into established systems.

5

#### BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 illustrates a contrast sensitivity function.

FIG. 2 illustrates an image processing technique.

FIG. 3 illustrates a coring function.

10

FIG. 4 illustrates another coring function.

FIG. 5 illustrates another coring function.

FIG. 6 illustrates a 1-dimensional filtering of a sharp edge with noise.

FIG. 7 illustrates graphs of filtering.

15

FIG. 8 illustrates graphs of non-linear low pass filter.

FIG. 9 illustrates a block diagram of a non-linear sieve low pass filter.

FIG. 10 illustrates a block diagram of a non-linear sieve low pass filter.

20

FIG. 11 illustrates a block diagram of a non-linear low pass filter.

#### DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

The present inventor observed that the many of the objectionable contouring effects in images appear in regions of the image that are generally free from a significant number of edges or otherwise high texture. The present inventor similarly observed fewer objectionable contouring effects in regions of the image that have a significant number of edges or otherwise high texture. After consideration of this difference in observing objectionable contouring effects, the present inventor considered that the human visual system has lower sensitivity to these contour effects in the high frequency regions of the image and

the human visual system has higher sensitivity to these contour effects in the low frequency regions of the image. FIG. 1 depicts the overall spatial response of the human visual system with its underlying visual channels with the coarse quantization below the threshold. Further, the masking effects by the higher frequency content of the image, which is limited to the channels as shown in FIG. 1, further inhibits the visibility of the steps in the waveforms in the high frequency regions of the image.

In order to preserve the quality of the image, the system preferably (albeit not necessary) reduces the effects of the objectionable contours without having to add noise to the image in order to hide them. Thus the system is suitable for use with images that are otherwise free of image capture noise, such as computer graphics and line art with gradients. Moreover, it would be preferable (albeit not necessary) that the technique is implemented without decision steps or if statements to achieve computational efficiency. Also, the technique should require not more than a single buffer of a size less than or equal to the image, and more preferable a buffer less than 30, 20, or 10 percent of the size of the image.

Referring to FIG. 2, an input image 100 is provided with bit-depth  $P$ . The bit-depth  $P$  is frequently 8-bits for many images. As previously described, the image with a bit depth of  $P$  is normally quantized into  $2^P$  different values but may exhibit contouring effects when displayed on a display with a different bit depth, such as a 10 bit display. In order to reduce those aspects of the image that are likely to exhibit contouring effects the present inventor came to the realization that the aspects of the image that will create contouring effects should be identified in a suitable manner. In another case, the image may be represented as  $P$  bits, but actually have quantization artifacts due to less than  $P$  bits. An example is a DVD image which is represented at 8 bits/color but only has 6 bits/color of real information because of the quantization of P or B-frames, or in the YCbCr to RGB conversion. In many cases, such as DVD applications, the bit-depth limitation comes from the inaccurate color matrix calculation (e.g., insufficient bit-depth in the registers).

To reduce the false step edges of contouring the image is preferably low-pass filtered 102. The low-pass filter applied to the image also acts to increase the bit depth, since the low pass filter is primarily an averaging operation (averaging across a neighborhood of pixels), and the averaging results in a higher precision. Other techniques may likewise be applied to effectively modify the bit depth of the image. In some cases, the system may also modify the bit depth by switching the number of bits used for processing the image, such as 8 bits to 16 bits. One way of characterizing an increase in the bit depth of an image is to modify in any manner an image to  $2^N$  different levels, where  $N \neq P$ . Alternatively, the image may already have a bit-depth needed/desired for the output ( $P=N$ ), but the image itself may have a limited bit depth of  $P$  ( $P < N$ ) from a previous operation. In this case, the value of  $P$  should be known or otherwise determined. Alternatively, the image may have a bit-depth of  $N$ , but the image itself may have a limited bit-depth of  $P$  ( $P < N$ ).

The result of the low pass filtering of the image is to modify the image to achieve a bit depth of  $N$ . In most cases the bit depth  $N$  is the desired bit depth in the final image, such as an image having bit depth  $N$  to be displayed on a  $N$  bit display. The low pass filter should be sufficiently wide (in the spatial domain) to reduce most false steps (contouring) due to the bit-depth limit  $P$ . It is noted that the low pass filter likewise reduces other desirable image information within the image, so merely low-pass filtering the image is insufficient. Note that the low pass filter also reduces much useful image information, such as by severely blurring the image, so this step is insufficient to properly rectify the undesirable contouring.

The low pass filter 102 may be implemented as Cartesian-separable 1-dimensional filters. The use of a pair of 1-dimensional filters is more computationally efficient than a 2-dimensional filter. The filter preferably has a rectangular shape. The size of the low pass filter kernel may be based upon the viewing distance and the pixels per inch of the display. For a 90 pixel per inch display viewed at 1000 pixel viewing distance, the kernel size should be more

than 11 pixels to remove the visibility of a contour edge, but the present inventor determined that the kernel size should actually be more than 31 pixels to remove the low frequency modulation component of the false contour. Accordingly, the kernel size should be based upon the low frequency modulation component of the false contour, as opposed to merely the visibility of the contour edge.

The resulting image from the low pass filter is primarily the low frequency components of the image. It is likewise to be understood that any suitable filter may be used, such as one that generally attenuates the high frequency components with respect to the low frequency components of the image.

The system may subtract the low pass filtered image 102 from the original bit-depth limited image 100 having false contours. This in essence reduces the low-frequency content of the input image, while primarily leaving the high frequency content. However, the subtraction 105 also results in another attribute, namely, the high frequency portion of the remaining image contains the high frequency portion of the contour artifacts. The result of this operation may be referred to as the HP component 107. It is also noted that the result of the low pass filter 102 is that the low frequency portion of the remaining image from low pass filter 102 contains the low frequency portion of the contour artifacts. Accordingly, the contouring artifacts are separated in some manner between the low frequency and high frequency components of the image.

The subtraction process 105 leads to a bit-depth increase of 1 due to the sign. For example a subtraction of two images with 0 to +255 results in an image with a range from -255 to +255. Hence the high pass component 107 has  $N+1$  bit depth (this is based upon a source image being padded to  $N$  bits before the subtraction operation). Padding may include inserting 0's and/or 1's to the least significant bits that are missing from the lower bit depth representations relative to the higher.

It is noted that the output of the system may not need  $N+1$  bits, but it should be able to carry a sign, and thus have a zero mean level. If only  $N$  bits are used and one bit is dedicated to the sign, then only  $N-1$  bits are available for

addition to the low pass filter image (in the last addition step). In that case some edge sharpness and detail may be lost.

As previously described, the result of subtracting the low pass filtered image from the original image results in an image that maintains high frequency false contour information. It has been determined that the high frequency false contour information that should be reduced are those having a low amplitude. Accordingly, the low amplitude information should be reduced with respect to the high amplitude information. To reduce the low amplitude high frequency false contour information a coring function 110 may be applied. Other techniques may likewise be used, if desired. The coring function 110 may include a hard-threshold coring function, such as for example, if  $\text{abs}(\text{HP}) < b$  then  $\text{HP} = 0$ , else  $\text{HP} = \text{HP}$ . This effectively reduces the contours, especially if the low pass filter is sufficiently large.

Unfortunately, simply applying a hard-threshold coring function, while acceptable, resulted in unexpected additional artifacts that appear like islands of color, and as ringing of step edges. After consideration of these unexpected artifacts, it was determined that a transitioned coring function will both reduce the low amplitude high frequency false contour information, and reduce the additional color islands and ringing of steps edges. A modified transitional coring function, may be for example, as:

$$\begin{aligned} (1) \quad \text{CVout} &= \text{sign}(\text{CVin}) * A[0.5 - 0.5 \cos(\alpha |\text{CVin}|)] && \text{for } |\text{CVin}| < M \\ \text{CVout} &= \text{CVin} && \text{for } |\text{CVin}| \geq M \end{aligned}$$

CVin is the input code value of the HP image to the coring function, while CVout is the output code value of the coring function. M is the merge point that is where the coring behavior ends (or substantially ends) and the coring function returns to the identify function (or other suitable function). A and  $\alpha$  are parameters selected to ensure two conditions at the merge point, M, namely:

- (A) amplitude = M
- (B) slope = 1



The first criterion ensures that the coring function has no discontinuity in actual value, and the second ensures that the 1<sup>st</sup> derivative is continuous. The first criterion keeps the tone scale monotonic in the HP band, and the second avoids mach band type artifacts. Thus this coring function could even be applied to the low pass band without such artifacts. It is noted that the coring function preferably has a slope that equals 1 that intersects with the origin of the plot as well as no second order discontinuities.

The criteria may be restated as follows:

$$(2) \quad M = A[0.5 - 0.5 \cos (\alpha M)] \quad \text{for the amplitude}$$

$$(3) \quad 1 = d/dC_{Vin} (A[0.5 - 0.5 \cos (\alpha C_{Vin})])|_{C_{Vin}=M}$$

An example of the coring functions for the value of  $M = 8$  with  $A = 11.3$  and  $\alpha=0.25$ , is shown in FIG. 3 to illustrate the actual mapping due to the effects of the quantization to  $N$  bits. Curve 50 has a slope equal to 1. The curve 52 is equation (1) with the parameters to achieve a merge point at 8. The curve 54 are the actual code values if the HP image is quantized to  $N$  bits. The curve 56 is the slope of the scaled cosine function in equation (1), before the merge point. It is noted that only the positive half of  $C_{Vin}$  is illustrated.

Another example of the coring function for the value of  $M=4$  with  $A=5.6$  and  $\alpha = 0.5$  is illustrated in FIG. 4. Another example of the coring function for the value of  $M=16$  with  $A=19.75$  and  $\alpha = 0.14$  is illustrated in FIG. 5.

Referring again to FIG. 2, the next step is to add the cored image 110 and the filtered low pass components 102 together at 112. This operation restores the low frequency information that was reduced back to the image. The result of this addition operation is  $N+1$  bits. It is  $N+1$  since the range may be larger than the input image (for example if the low pass component for  $N$  bits may be 0 to +255, the high pass component may have a range of  $N+1$  (-255 to +255) and the result is -255 to +512). Anything out of the range of  $N$  bits is clipped (e.g., out of range 0 to +255 for  $N=8$ ). It turns out that there are a limited number of pixels

that fall out of that range, and when they do, they are usually isolated edge pixels. The clipping that occurs as a result is not readily visible in the final image 114.

The number of gray levels is given by N or P, which are bit-depths, so the numbers of levels is  $2^N$  or  $2^P$ . However, another embodiment is to not use bit-depth to determine the number of gray levels, but the numbers of gray levels directly.

In a particular implementation, each of the steps shown in FIG. 2 may be applied to the entire image in a sequential manner, which may result in the need for large buffers. However, a more memory efficient technique involves using as a sliding window, where the computations within the window are used to compute the pixel at the center of the window (or in some other position in the window). This reduces the memory requirements.

In a typical implementation the bit-depth of the input and output images are known in advance. This is typically the case in many applications, such as display an eight bit image on a 10-bit display.

While the implementation illustrated in FIG. 2 effective reduces the contour artifacts, it was determined that it also results in ghost artifacts in some areas of some images. Upon further analysis the present inventors came to the realization that the ghost artifacts occur in regions with sufficiently sharp edge features. While such a result is satisfactory for most cases because the viewer is unaware of exactly how the image should really appear, it turns out to be particularly detrimental in the facial region of an image, where there is a higher expectation of the appropriate texture by the typical viewer.

To reduce the occurrence of ghosting artifacts, the present inventors came to the realization that a linear low pass filter tends to smear sharp edges because a sharp edge has high energy on the high-pass frequency band and this energy does not efficiently pass through the low pass filter. When a pixel with value  $I$  is close to a sharp edge, the output of the low pass filter, the low pass component  $I_L$ , is quite different from  $I$ . In other words, the high pass component  $I_H = I - I_L$  is substantial. Because the selected coring functions passes its input signal without

modification to its output if an input-signal  $I_H$  is larger than a threshold, the substantial  $I_H$  is unchanged after coring and consequently  $I$  is unchanged. As a result, the decontouring technique using a linear low pass filter maintains areas close to sharp edges unchanged. With noise in other areas is reduced, the noise in the areas close to sharp edges become prominent. In other words, the noise in regions not sufficiently proximate the sufficiently sharp edges is reduced to a greater extent than the noise in regions sufficiently proximate the sharp edges. This phenomena is further illustrated by FIGS. 6 and 7.

FIG. 6 illustrates a 1-dimensional signal with a sharp edge and some noise. FIG. 7(a) illustrates the results of linear low pass filter being applied to this signal. The sharp edge in FIG. 6 becomes a slope in FIG. 7(a). FIG. 7(b) illustrates the results of high pass filtering. In the region around the sharp edge, there are two substantial spikes. The widths of the spikes are typically equal to the length of the filter, which in this illustrate is 31. FIG. 7(c) illustrates the high pass filter signals after coring. The two spikes remain unchanged while other fluctuations are significantly smooth by the coring function. FIG. 7(d) illustrates the final result by adding the low pass filter and the cored high pass filter. As it may be observed, there remain ghost effects around the sufficiently sharp edges. Accordingly, it is desirable to reduce or otherwise eliminate the energy remaining near an edge to substantially the same level as the energy remaining in regions not near an edge.

A modified low pass filter may be used to preserve sharp edges. For example, a Sieve filter may be designed to be a low pass filter while preserving sharp edges. The sieve filter utilizes a 1-dimensional or 2-dimensional rectangular or other suitably shaped window, where the current pixel  $I(x,y)$  is at the center of the window. The filter may compare all the pixels  $I(i,j)$  in the window with the central pixel  $I(x,y)$ . The filter may average those pixels whose value differences with the central pixel  $I(x,y)$  is within a threshold  $T$ . Because this filter drops pixels that are not within the threshold, one may refer to this type of filter as a sieve filter. Because a sieve filter does not satisfy the conditions of linear filter, it

is referred to as a nonlinear filter. Mathematically, the output of the sieve filter,  $I_{LP}(x,y)$ , is calculated by

$$I_{LP}(x, y) = \frac{\sum_{(i,j) \in E \& |I(i,j)-I(x,y)| < T} I(i, j)}{N(x, y)} \quad (1)$$

where  $E$  is the window;  $N(x,y)$  is the count of the pixels in  $E$  that satisfy the condition of  $|I(i,j)-I(x,y)| < T$ . In some cases, a sieve filter may be incorporated with any de-contouring technique or otherwise an artifact reduction technique since the filter tends to estimate the desirable signal from its noisy observation.

FIG. 8 show that the same signal in FIG. 6 processed by the sieve filter. Figure 5(a) shows the results of non-linear sieve low pass filter applied to this signal. When the threshold  $T$  is set to be 16, the sharp edge is preserved, while FIG. 7(a) is a slope. FIG. 8(b) shows the results of high pass filter. Contrast to FIG. 7(b), in the area around the sharp edge, there are no spikes. FIG. 8(c) shows the high pass filter signals after coring. All substantial fluctuations are removed by the coring function. FIG. 8(d) shows the final result by adding low pass filter and the cored high pass filter. There are no ghost effects around the sharp edge.

A block-diagram of an exemplary sieve filter is shown in FIG. 9.

One technique to determine the threshold is by presetting or learning based on the local information. Learning may use histogram of the window.

The sieve filter mathematically defined by Equation (1) and illustrated by the diagram of FIG. 9 is problematic to implement by the hardware. Therefore, an alternative version of the sieve filter may defined by the following equation

$$I_{LP}(x, y) = \frac{\sum_{(i,j) \in E \& |I(i,j)-I(x,y)| < T} I(i, j) \times (N_{total}N(x, y)) \times I(x, y)}{N_{total}} \quad (2)$$

When a pixel's difference with the central pixel is bigger than the threshold  $T$ , instead of dropping the pixel from summation in Equation (1), Equation (2) replaces the pixel's value with the central pixel's value in summation. Equation (2) has the advantage over Equation (1) that the

denominator in Equation (2) is fixed, which is preferable by hardware implementation. The corresponding diagram is shown in FIG 10.

The implementation hardware block-diagram using a sieve filter (horizontal 31 taps and vertical 5 taps) is shown in FIG. 11. The diagram utilizes Equation (2). The sieve filter is generally not Cartesian-separable. To simplify the requirement for calculation, one may apply the sieve filter horizontally. Specifically, as shown in FIG. 10, the technique may average the vertical pixels first, and then apply Equation (2) to the 31 averaged data horizontally.

As it may be observed the low pass filter may be any non-linear low pass filter, as desired. In addition, the low pass filter may be any spatially-varying low pass filter, as desired. Also, the low pass filter may be any spatially-varying non-linear low pass filter.

It is noted that if the system has access to the image and it is desirable to reduce the bit-depth in subsequent image processing, such as display on a lower bit depth liquid crystal display, while maintaining a high image quality for the reduced bit depth, then a particular class of image processing techniques may be used, such as for example, dithering techniques. Several different types of dithering techniques have been developed, such as for example, amplitude dithering, spatial dithering, and phase dithering. For example, a properly designed amplitude dither technique may preserve the image entropy at low frequencies, while allowing for entropy losses at high frequencies.

However, if the image that is acquired already has a lower bit-depth than desired, and a dithering technique was not previously applied to the image, then the dithering techniques do not assist in removing the undesirable distortion. In other words, dithering techniques are traditionally used for pre-processing of images. The reason for this pre-processing limitation is because the entropy loss has already occurred at the low frequencies due to quantization, which acts to reduce the amplitude resolution uniformly across all frequencies.

While the described implementations effectively reduce the contour artifacts, it was determined that it also removes a significant amount of low amplitude detail. While such a result is satisfactory for most cases because the

viewer is unaware of exactly how the image should really appear, it turns out to be particularly detrimental in the facial region of an image, where there is a higher expectation of the appropriate texture by the typical viewer.

To reduce the loss of low amplitude texture detailed, the present inventors came to the realization that one may separate the case where the low amplitude information is due to the image texture (which is generally isotropic) versus the case when the low amplitude information is due to a false contour (which is generally non-isotropic, a.k.a., structured error). Generally isotropic information (“iso” meaning one or the same, and “tropic” meaning space or direction) has generally the same texture in all directions. For example, clean sand on the beach may be considered generally isotropic. For example, generally non-isotropic information has different textures in different directions. In many cases, the non-isotropic information has the characteristic of an edge. Accordingly, generally isotropic information has more uniform texture relative to generally non-isotropic information. This additional filtering may be included with the coring.

While a plurality of different filters may be used, it is preferable to use a single filter, which is more computationally efficient in many cases. The accumulation of local special activity may be determined in a suitable manner, such as for example, a single sigma type filter or a single sum of absolute differences (“SAD”) type filter.

The preferred system uses a version where the lowpass filter is implemented in two Cartesian separable steps (i.e., a cascade of H and V steps.), and the filter width spatially is = 31 pixels, with a uniform (rectangular) impulse response. For many images a filter width of 17 or 25 is acceptable. Typically a width less than 17 does not work exceptionally well for image of size greater than 512 x 512.

The system may accumulate local activity, if desired, over a local region around a pixel of the image to be cored (examples include standard deviation,  $\alpha$ , over a local 9x9 window, or SAD, over a similar local window). One may select different coring functions based on a local activity index (or continuously adjust

coring function based on a local activity index). Local activity measure could be taken from the mean of the local window, but is more preferably around the zero activity point of the HP image (i.e., 0, unless a pedestal offset is used.). The  
5       coring function has merge point where coring reverts to an identity function. At this merge point, the slope of the coring equals the slope of the identity function (typically = 1).

The processing of the image may result in an image having a bit depth greater than the desired output bit-depth “N” for the image. Accordingly, the bit  
10       depth of the input image may have been the same as, or less than, the desired bit-depth of the output image which during the image processing technique for reducing the contouring effects is increased to a bit-depth greater than the desired bit-depth of the output image. The present inventors came to the realization that with an image having a greater bit depth than necessary for the output image, as a  
15       result of previous image processing for de-contouring effects (e.g., low pass filtering), there is the potential that a dithering technique (which may be included together with the coring) (e.g., bit depth extension) may be applied to the image when reducing the bit depth to the desired output bit depth. As previously  
discussed, dithering techniques are typically not considered applicable for  
20       overcoming bit-depth limitations for image processing when the image provided has a bit depth less than the bit depth of the output image. The image may be modified with a dithering technique, such as for example, amplitude dithering, spatial dithering, and phase dithering. The dithering technique may allow the bit depth to be reduced without noticeable visible loss from R+1 to N. Preferably, an  
25       amplitude based dither pattern is applied that is characterized by the frequency spectrum that is approximately matched to the inverse of the human visual spatial frequency response.

Another technique that may be applied is to increase the bit depth of the high pass component 107 to a greater bit depth, such as “R”. In this manner, a  
30       more accurate coring function may be applied in the R-bit space. In some implementations, this may be accomplished by modifying the bit depth of the

input image to R bits before the subtraction that form the high pass component  
107.

The dithering technique may be applied to the low pass filtered image, if  
desired. This reduces the effects of contouring effects as a result of quantization  
5 while similarly increasing the quality of the output image.

The dithering technique may be applied to the input image by first  
expanding the bit depth of the image. While spectrally shaped “noise” may be  
used, in addition “white” noise may likewise be used, if desired. The low pass  
filter will effectively modify the “white” noise to a “high pass” noise, which has  
10 the general desired spectral characteristics.